RECEIVED
CENTRAL FAX CENTER
SEP 15 2006

IN THE CLAIMS

1. (Currently Amended) A method of initializing a plurality of processors in an integrated circuit, the method comprising the steps of:

identifying each one of the processors;

executing boot code for initializing each one of the processors, the boot code containing ~~specific~~ unique instructions ~~code~~ for at least one of the processors and a single set of instructions that is used for the initialization of ~~common code that is common for~~ each one of the processors, the ~~specific code~~ unique instructions being accessed according to the identity of the processor executing the boot code.

2. (Original) The method of claim 1 wherein the step of identifying includes the step of:

assigning a unique value to each one of the processors.

3. (Original) The method of claim 2 wherein each one of the processors includes a register and the step of assigning includes the step of:

providing a unique value to the register of the processor.

4. (Currently Amended) An apparatus for initializing a plurality of processors in an integrated circuit, the apparatus comprising:

means for identifying each one of the processors;

means for executing boot code for initializing each one of the processors, the boot code containing ~~specific code~~ unique instructions for at least one of the processors and a single set of instructions~~common code that is common~~ that is used for the initialization for each one of the processors, the ~~specific code~~ unique instructions being accessed according to the identity of the processor executing the boot code.

5. (Original) The apparatus of claim 4 wherein the means for identifying includes:

means for assigning a unique value to each one of the processors.

6. (Original) The apparatus of claim 5 wherein each one of the processors includes a register and the means for assigning includes:

means for providing a unique value to the register of the processor.

7. (Currently Amended) An integrated circuit comprising:

a plurality of processors each having a unique identifier;

a bus for providing data to and from each one of the processors;

a memory, coupled to the bus, having a set of instructions that is used for the initialization of ~~boot code for initializing~~ each one of the processors and unique instructions for each one of the processors, the boot code using the unique identifier of each processor to access instructions ~~code~~ that is unique to the identified processor.

8. (Original) The integrated circuit of claim 7 wherein each one of the processors has at least one register, and the at least one register is used to store a unique identifier.

9. (Original) The integrated circuit of claim 7 further comprising a cache shared between the plurality of processors.

10. (Currently Amended) A method of initializing multiple processors in an integrated circuit, the method comprising the steps of:

creating boot code for initializing the processors, the boot code having a single set of instructions~~code~~ that is used~~common~~ for each one of the processors, and unique instructions ~~code~~ that ~~is~~are specific for each one of the processors;

assigning each one of the processors a unique identifier; and

executing the boot code and accessing the unique instructions~~code~~ that ~~is~~are specific for a processor using the unique identifier of the processor.

11. (Original) The method of claim 10 wherein the step of assigning includes the step of:

storing a unique value into a register of each one of the processors.

Please add the following new claims:

12. (New) The method of claim 3 wherein the step of executing the boot code includes the steps of:

accessing the same memory location for each one the processors when executing the single set of instructions; and

accessing a memory location that is different from that used to store the single set of instructions for the unique instructions.

13. (New) The apparatus of claim 4 wherein the means for executing the boot code includes:

means for accessing the same memory location for each one the processors when executing the single set of instructions; and

means for accessing a memory location that is different from that used to store the single set of instructions for the unique instructions.

REJECTION OF CLAIMS 1-11

In the current Office Action, the Examiner rejected claims 1-8 and 10-11 under 35 U.S.C. section 102(e) as being anticipated by U.S. Patent No. 6,701,429 to Gustafsson et al. ("Gustafsson") and claim 9 as being obvious over Gustafsson in view of U.S. Patent No. 5,895,487 to Boyd et al. ("Boyd"). The cited references are discussed below.

Gustafsson

Gustafsson discloses a multi-processor system that uses either a single of multiple "WhoAMI register(s) to start-up each processor (Col. 5 lines 56-64). The WhoAMI register contains an identification number associated with that processor (Col 5 lines 64-

66). The system includes a "jumpstation" having unique instructions for each one of the processors (see Figure 3 and associated explanation Col. 6 lines 31-67 and Col. 7 lines 1-36).

Gustafsson fails, however, to disclose a single set of common instructions that are used to initialize each one of the processors. In fact, Gustafsson specifically uses a separate location for storing instructions for each one of the processors

In contrast, Applicant's present invention, as now defined by the pending claims, includes a single set of instructions for initializing all of the processors and then unique instructions for at least one of the processors where the identification of the processor is used to retrieve the unique instructions.

Boyd

The Examiner cited Boyd for disclosing a shared cache.